



Original software publication

# Xcompact3D: An open-source framework for solving turbulence problems on a Cartesian mesh



Paul Bartholomew<sup>a</sup>, Georgios Deskos<sup>b</sup>, Ricardo A.S. Frantz<sup>c</sup>, Felipe N. Schuch<sup>c</sup>,  
Eric Lamballais<sup>d</sup>, Sylvain Laizet<sup>a,\*</sup>

<sup>a</sup> Department of Aeronautics, Imperial College London, London, SW7 2AZ, UK

<sup>b</sup> National Renewable Energy Laboratory, Golden, CO 80401, USA

<sup>c</sup> School of Technology, Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil

<sup>d</sup> Université de Poitiers, CNRS, ISAE-ENSMA, Institut Pprime, F-86962 Futuroscope Chasseneuil, France

## ARTICLE INFO

### Article history:

Received 22 November 2019

Received in revised form 25 February 2020

Accepted 12 June 2020

### Keywords:

High-order finite-difference

Turbulence

Cartesian mesh

High Performance Computing

Computational Fluid Dynamics

## ABSTRACT

Xcompact3D is a Fortran 90–95 open-source framework designed for fast and accurate simulations of turbulent flows, targeting CPU-based supercomputers. It is an evolution of the flow solver Incompact3D which was initially designed in France in the mid-90's for serial processors to solve the incompressible Navier–Stokes equations. Incompact3D was then ported to parallel High Performance Computing (HPC) systems in the early 2010's. Very recently the capabilities of Incompact3D have been extended so that it can now tackle more flow regimes (from incompressible flows to compressible flows at low Mach numbers), resulting in the design of a new user-friendly framework called Xcompact3D. The present manuscript presents an overview of Xcompact3D with a particular focus on its functionalities, its ready-to-run simulations and a few case studies to demonstrate its impact.

© 2020 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## Code metadata

Current code version	3.0
Permanent link to code/repository used for this code version	<a href="https://github.com/ElsevierSoftwareX/SOFTX_2019_357">https://github.com/ElsevierSoftwareX/SOFTX_2019_357</a>
Legal Code License	GNU General Public License (GPLv3)
Code versioning system used	git
Software code languages, tools, and services used	Fortran90, MPI
Compilation requirements, operating environments & dependencies	GNU Fortran, Intel Fortran Compiler, Cray Fortran Compiler, PGI Fortran Compiler
If available Link to developer documentation/manual	
Support email for questions	<a href="mailto:s.laizet@imperial.ac.uk">s.laizet@imperial.ac.uk</a>

## 1. Motivation and significance

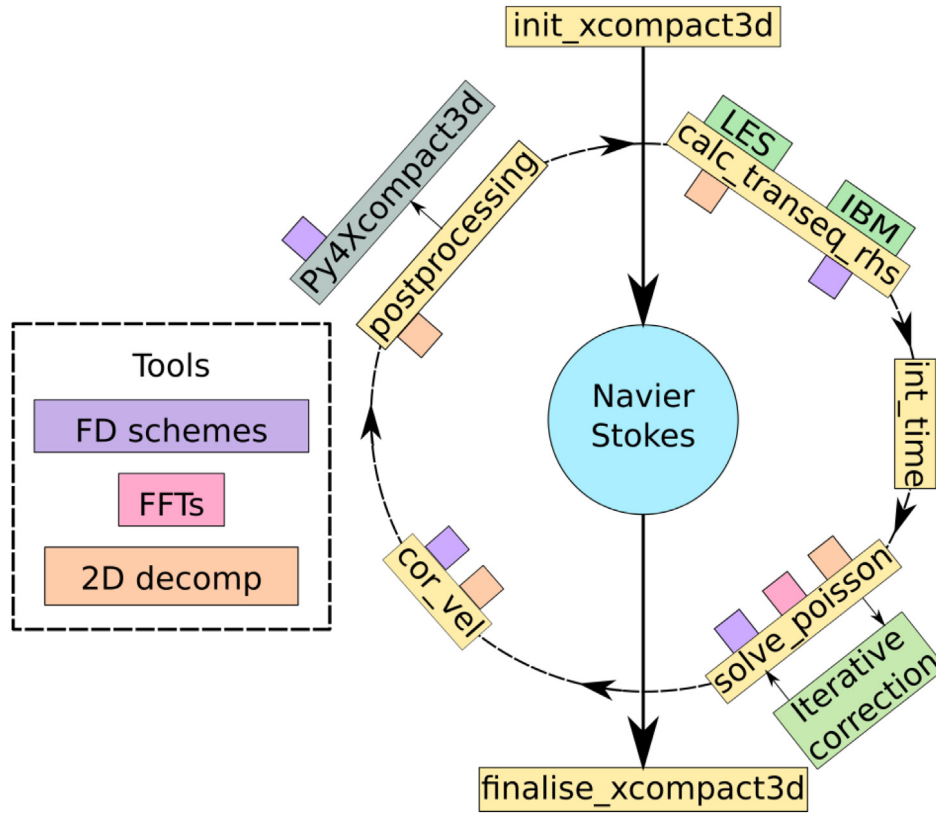
Understanding, predicting and controlling turbulent flows is of central importance and a limiting factor to a vast range of industries and scientific endeavours. It remains one of the most challenging problems in science. Furthermore, the design of many engineering and industrial systems as well as the prediction of

their impact on the environment, greatly relies on the turbulent behaviour of fluid flows being properly quantified.

While the Navier–Stokes equations constitute a broadly accepted mathematical model to describe the motions of a turbulent flow, their solutions can be extremely challenging to obtain due to the chaotic and inherently multi-scale nature of turbulence. Turbulent scales typically cover several orders of magnitude, requiring significant computing power and memory to resolve. Thanks to the recent dramatic increase in computational resources, high-fidelity simulations of turbulent flows, namely Direct and Large Eddy Simulations (DNS/LES), for which

\* Corresponding author.

E-mail address: [s.laizet@imperial.ac.uk](mailto:s.laizet@imperial.ac.uk) (S. Laizet).



**Fig. 1.** Diagram of the Xcompact3D software architecture, a yellow background indicates the major functionalities of Xcompact3D. Optional functions are indicated in green. Also shown is the external postprocessing tool Py4Incompact3d. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the vast majority of the turbulent scales are resolved, have become a widely used technique for the prediction of turbulent flows and Computational Fluid Dynamics (CFD) is now a critical complement to experiments and theory.

The implementation of a numerical code for high-fidelity simulations of turbulent flows requires a wide range of expertise including both numerical methods and computer science. Correctly setting up a simulation for a given flow requires similar expertise. A major motivation of the Xcompact3D framework is the provision of a tool which can be used “off-the-shelf” to obtain high-quality simulations of turbulent flows. Further motivating the development of Xcompact3D (*né* Incompact3D [1,2]) is to bring together the numerous versions of Incompact3D developed in the last few years and to add new capabilities to the solver while making it more user friendly and flexible.

As a Cartesian mesh solver, Xcompact3D has a well-defined *raison d'être* and mission within the CFD community. Its unique features are its accuracy, efficiency, scalability, independence of external libraries apart from a MPI library, user-friendliness and its customisability. In particular, ready-to-run simulations are provided for users with limited experience in CFD as well as a template for the only input file needed to run a simulation in a new set-up. Compilation is very fast and simple with only one Makefile and no dependence on external libraries (except for MPI). The numerical parameters of a simulation (domain size, number of mesh nodes, number of cores, etc.) can easily be changed without having to recompile the code. The manuscript focuses on exactly these aspects. Section 2 is dedicated to the software architecture of Xcompact3D including its parallel computing strategy, functionalities, main features and scalability. Subsequently, illustrative examples and impact case studies are shown in Section 3 and Section 4. Finally conclusions and future directions are briefly discussed in Section 5.

## 2. Software description

The turbulent flows described in Xcompact3D are governed by the Navier–Stokes equations, given for low Mach number (LMN) flows as

$$\frac{D\rho}{Dt} = -\rho \frac{\partial u_i}{\partial x_i} \quad (1)$$

$$\frac{\partial p^{(0)}}{\partial x_j} = 0 \quad (2)$$

$$\rho \frac{DT}{Dt} = \frac{1}{Re} \frac{\partial}{\partial x_j} \kappa \frac{\partial T}{\partial x_j} \quad (3)$$

$$\rho \frac{Du_i}{Dt} = -\frac{\partial p^{(1)}}{\partial x_i} + \frac{1}{Re} \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i \quad (4)$$

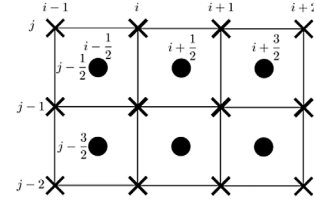
$$p^{(0)} = \rho T \quad (5)$$

where  $u_i$  is the velocity vector,  $p$  the pressure,  $\rho$  the density,  $T$  the temperature,  $\tau_{ij}$  the viscous stress tensor,  $\kappa$  the heat diffusion coefficient and  $D(\cdot)/Dt$  is the material derivative.  $Re$  is the Reynolds number, a dimensionless number defined as the ratio between inertial and viscous forces, which determines the range of turbulent scales to simulate. Similarly, the Prandtl number  $Pr$  is a dimensionless number defined as the ratio of momentum to thermal diffusivity. As shown in (1)–(5) under the LMN approximation the pressure field may be decomposed into the thermodynamic  $p^{(0)}$ , and mechanical  $p^{(1)}$  pressures. The former is constant in space which allows filtering of the acoustic pressure waves from the solution, whilst the mechanical pressure behaves as the pressure field of an incompressible flow; for further details the interested reader is referred to [3–5]. This formulation allows Xcompact3D to tackle flows with variable density in the regime  $0 \leq M \lesssim 0.3$

**Table 1**

Table of discrete operators (differentiation, interpolation and filtering) with the coefficients for second and sixth order schemes.

Operator	Compact scheme	$\mathcal{O}(6)$ scheme	$\mathcal{O}(2)$ scheme	$\mathcal{O}(6)$ hyper-viscous scheme
First Derivative	$\alpha f'_{i-1} + f'_i + \alpha f'_{i+1} = a \frac{f_{i+1} - f_{i-1}}{2\Delta x} + b \frac{f_{i+2} - f_{i-2}}{4\Delta x}$	$\alpha = 1/3,$ $a = 14/9,$ $b = 1/9$	$\alpha = 0,$ $a = 1,$ $b = 0$	-
Second Derivative	$\alpha f''_{i-1} + f''_i + \alpha f''_{i+1} = a \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + b \frac{f_{i+2} - 2f_i + f_{i-2}}{4\Delta x^2} + c \frac{f_{i+3} - 2f_i + f_{i-3}}{9\Delta x^2} + d \frac{f_{i+4} - 2f_i + f_{i-4}}{16\Delta x^2}$	$\alpha = 2/11,$ $a = 12/11,$ $b = 3/11,$ $c = 0,$ $d = 0$	$\alpha = 0,$ $a = 1,$ $b = 0,$ $c = 0,$ $d = 0$	$\alpha = \frac{1}{2} - \frac{320k''_m \Delta x^2 - 1296}{405k''_c \Delta x^2 - 640k''_m \Delta x^2 + 144},$ $a = -\frac{4329k''_c \Delta x^2 / 8 - 32k''_m \Delta x^2 - 140k''_c \Delta x^2 k''_m \Delta x^2 + 286}{405k''_c \Delta x^2 - 640k''_m \Delta x^2 + 144},$ $b = \frac{2115k''_c \Delta x^2 - 1792k''_m \Delta x^2 - 280k''_c \Delta x^2 k''_m \Delta x^2 + 1328}{405k''_c \Delta x^2 - 640k''_m \Delta x^2 + 144},$ $c = -\frac{7695k''_c \Delta x^2 / 8 + 288k''_m \Delta x^2 - 180k''_c \Delta x^2 k''_m \Delta x^2 - 2574}{405k''_c \Delta x^2 - 640k''_m \Delta x^2 + 144},$ $d = \frac{198k''_c \Delta x^2 + 128k''_m \Delta x^2 - 40k''_c \Delta x^2 k''_m \Delta x^2 - 736}{405k''_c \Delta x^2 - 640k''_m \Delta x^2 + 144},$
First Derivative (interpolatory)	$\alpha f'_{i-1/2} + f'_{i+1/2} + \alpha f'_{i+3/2} = a \frac{f_{i+1} - f_i}{\Delta x} + b \frac{f_{i+2} - f_{i-1}}{3\Delta x}$	$\alpha = 9/62,$ $a = 63/62,$ $b = 17/62$	$\alpha = 0,$ $a = 1,$ $b = 0$	-
Interpolation	$\alpha f^I_{i-1/2} + f^I_{i+1/2} + \alpha f^I_{i+3/2} = a \frac{f_{i+1} + f_i}{2} + b \frac{f_{i+2} + f_{i-1}}{2}$	$\alpha = 3/10,$ $a = 3/4,$ $b = 1/20$	$\alpha = 0,$ $a = 1,$ $b = 0$	-
Interpolation (optimised)	$\alpha f^I_{i-1/2} + f^I_{i+1/2} + \alpha f^I_{i+3/2} = a \frac{f_{i+1} + f_i}{2} + b \frac{f_{i+2} + f_{i-1}}{2} + c \frac{f_{i+3} + f_{i-2}}{2} + d \frac{f_{i+4} + f_{i-3}}{2}$	$\alpha = 0.461658,$ $a = (75 + 70\alpha - 640d)/128,$ $b = (126\alpha - 25 + 2304d)/256,$ $c = (3 - 10\alpha - 1280d)/256,$ $d = 0.00146508$		
Interpolation (aggressive)	$\alpha f^I_{i-1/2} + f^I_{i+1/2} + \alpha f^I_{i+3/2} = a \frac{f_{i+1} + f_i}{2} + b \frac{f_{i+2} + f_{i-1}}{2} + c \frac{f_{i+3} + f_{i-2}}{2}$	$\alpha = 0.49,$ $a = (75 + 70\alpha)/128,$ $b = (126\alpha - 25)/256,$ $c = (3 - 10\alpha)/256$		
Filtering	$\alpha \hat{f}_{i-1} + \hat{f}_i + \alpha \hat{f}_{i+1} = a f_i + \frac{b}{2}(f_{i+1} + f_{i-1}) + \frac{c}{2}(f_{i+2} + f_{i-2}) + \frac{d}{2}(f_{i+3} + f_{i-3}).$	$a = \frac{1}{16}(11 + 10\alpha),$ $b = \frac{1}{32}(15 + 34\alpha),$ $c = \frac{1}{16}(-3 + 6\alpha),$ $d = \frac{1}{32}(1 - 2\alpha)$		



where  $M$  is the Mach number. In the case of constant density it reduces to the governing equations for incompressible flows. In addition, Xcompact3D supports simulations with multiple scalar fields via the following equation

$$\rho \frac{D\phi_s}{Dt} = \frac{1}{ReSc_s} \frac{\partial}{\partial x_i} \mathcal{D}_s \frac{\partial \phi_s}{\partial x_i} \quad (6)$$

where  $\phi$  is a scalar,  $\mathcal{D}$  a scalar diffusion coefficient,  $Sc$  the Schmidt number defined as the ratio of kinematic viscosity to mass diffusivity and subscript  $s$  is the scalar counter.

## 2.1. Software architecture

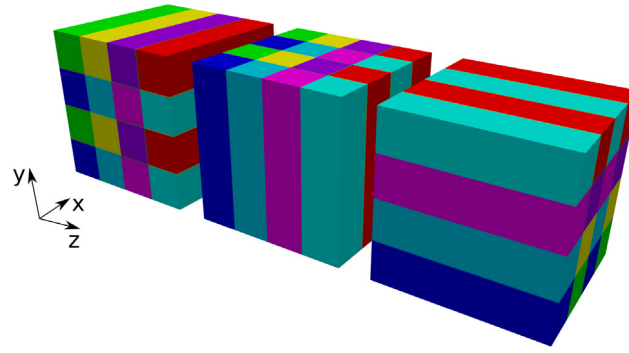
An overview of the modular structure of the framework is presented in Fig. 1. Xcompact3D is build around the 2D Decomp & FFT library [2,6] which is designed to implement a 2D pencil decomposition (as shown in Fig. 2(a)) for data distribution on distributed-memory platforms. This library also provides a highly scalable and efficient interface to perform three-dimensional distributed Fast Fourier Transforms, which are used to solve the Poisson equation (needed in the Navier–Stokes equations subjected to LMN or incompressibility approximations). Equations 1 – 5 are solved with high-order compact finite-difference schemes on a Cartesian mesh, with a conventional fractional step strategy based on the general principle of projection methods [7,8]. Since DNS of high Reynolds number flows is very often not possible due to its computational cost, LES can be performed with an implicit approach for which the sub-grid scale (SGS) stress tensor is modelled using numerical dissipation [9,10]. Different explicit SGS models are also available for

completeness, namely the standard Smagorinsky model and its so-called dynamic version (see [11] and [12] for more details). For flows around relatively complex geometries, it is possible to use a customised Immersed Boundary Method (IBM) [13]. IBMs, originally proposed by [14] are very attractive for Cartesian meshes, avoiding sophisticated body-fitted approaches and their associated loss of simplicity and increase of computational cost. Finally, please note that there exists a new Python library for the post-processing of the data, Py4Incompact3d, based on the same schemes as in Xcompact3D.<sup>1</sup>

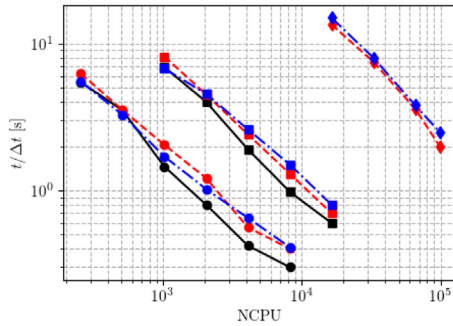
## 2.2. High-order finite-difference schemes and spectral Poisson solver

The superior properties of high-order schemes for DNS/LES in comparison to more conventional low-order schemes is now fully recognised, particularly regarding their ability to accurately capture a wider range of turbulent scales for a given spatial resolution. Standard spectral methods based on Fourier or Chebyshev representation yield highly accurate and efficient solutions to the Navier–Stokes equations, however with severe restrictions to their applicability. High-order compact finite-difference schemes on the other hand, approach the accuracy of spectral methods and allow greater flexibility in the selection of boundary conditions (in Xcompact3D, it is possible to use periodic, Dirichlet and Neumann boundary conditions). Although compact schemes are implicit in space, they are very competitive in terms of computational efficiency by comparison to more advanced spectral

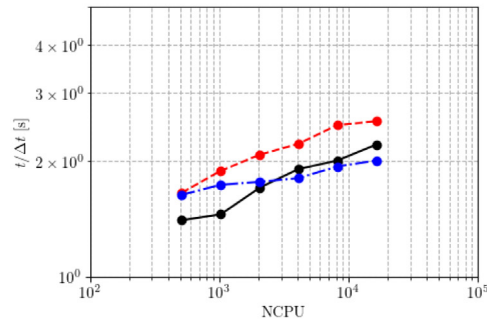
<sup>1</sup> This framework can be found [here](https://github.com/Py4Incompact3d/Py4Incompact3d).



(a) The 2D pencil decomposition, pencils coloured by MPI rank, representing  $x$  pencils,  $y$  pencils and  $z$  pencils, from left to right, respectively.



(b) Strong scaling behaviour.



(c) Weak scaling behaviour.

**Fig. 2.** The pencil decomposition and scaling behaviour of Xcompact3D with  $1025 \times 1025 \times 1025$  (circle),  $4097 \times 1025 \times 1025$  (square) and  $4097 \times 4097 \times 4097$  (diamond) mesh nodes for the strong scaling; 1 million mesh nodes per core are used for the weak scaling. Colour corresponds to computer: black – MareNostrum, red – ARCHER, blue – HAZEL HEN; dotted black lines show ideal scaling behaviour. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

element methods designed for body-fitted meshes (see [15,16] for more details).

Table 1 presents an overview of the compact schemes used in Xcompact3D. The velocity field information is on the main mesh (cross nodes) while the pressure field on the half-staggered one (dots), see inset diagram in Table 1. Various second derivative schemes are available, given in Table 1, depending on the type of simulation (hyper-viscous scheme for ILES, other schemes for DNS and explicit LES). The filtering scheme is only for use in explicit LES for the dynamic Smagorinsky model. Various sixth-order interpolation schemes with different properties for the small scales are available, but the recommendation is to use the optimised scheme as it can improve the quality of the solution in the context of LES. Note that the more conventional second-order schemes are also presented for completeness.

Xcompact3D advances the governing equations via the fractional step method [7,8], forming a Poisson equation for the pressure by taking the divergence of the momentum equations. This operation employs the interpolatory first derivatives given in Table 1 so that pressure is evaluated at the centre of control volumes (circle nodes in inset diagram) where the incompressibility condition is checked, the pressure gradients to correct the velocity are then evaluated by the reverse operation; this partial staggering is done to prevent spurious oscillations [1]. One of the main originalities of Xcompact3D is that the Poisson equation is solved in spectral space using the concept of modified wave numbers [17] for which operations in physical space are strictly equivalent to those in spectral space. Such a direct strategy, to avoid the use of expensive iterative techniques, is not new for

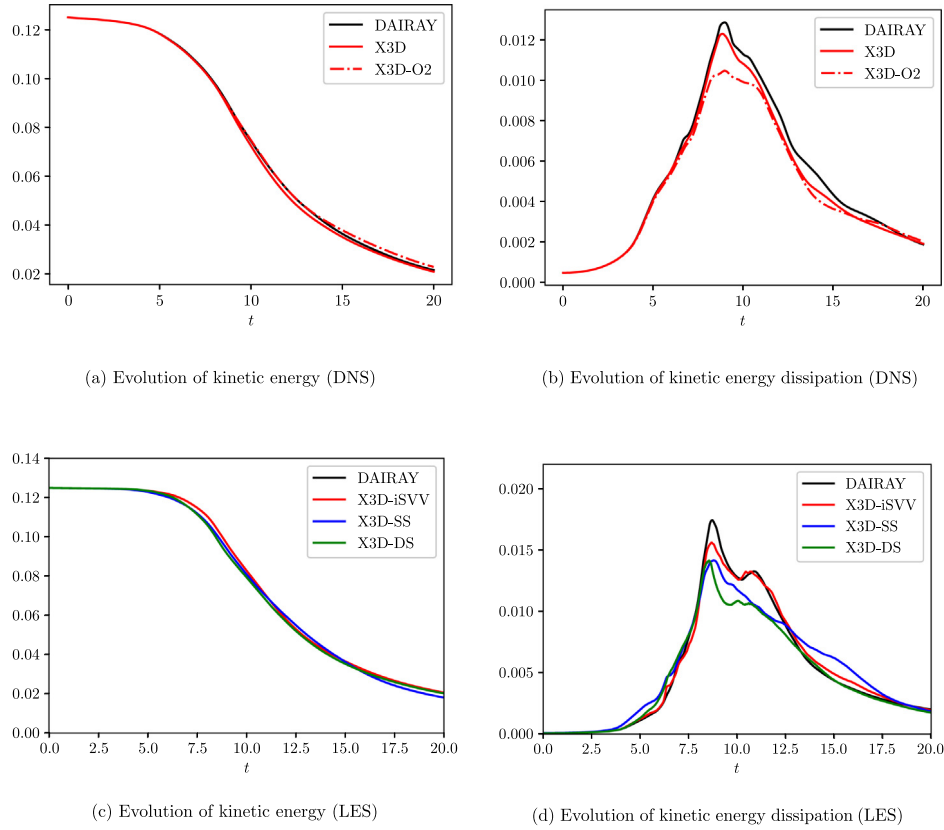
periodic and/or free-slip boundary conditions for the velocity field [18], however it has not been used in the past for Dirichlet boundary conditions combined with high-order finite-difference schemes [1].

### 2.3. Time advancement

The flow field is initialised either with an initial condition (`init_xcompact3d` in Fig. 1) or by loading a restart file. The Navier–Stokes equations advance in time through the fractional step method, implemented in 4 key steps. First, the right hand side terms are evaluated numerically, `calc_transeq_rhs` in Fig. 1 and integrated in time (with e.g. Runge–Kutta or Adams–Bashforth schemes), generating an intermediate velocity field, `int_time` in Fig. 1. The pressure field is then evaluated by solving the Poisson equation, obtained from the calculation of the divergence of the intermediate velocity field, equal to the divergence of the gradient of the pressure field, `solve_Poisson` in Fig. 1. Finally, the updated velocity field is computed by adding the pressure gradient to the intermediate velocity field, `cor_vel` in Fig. 1. At the end of a time step, the user can decide which quantities to store (`postprocessing` in Fig. 1).

### 2.4. The 2D Decomp & FFT library

The finite-difference schemes and spectral Poisson solver employed by Xcompact3D naturally break down into a series of 1D subproblems. It is therefore natural to parallelise the computational domain with a pencil decomposition as illustrated



**Fig. 3.** Results from Taylor–Green Vortex simulations at  $Re = 1,600$  (top row, with reference data of [10] in black) in a DNS setup and for  $Re = 10,000$  (bottom row, with reference DNS data of [10]). In the figures X3D-O2 corresponds to DNS data produced with conventional second-order schemes, X3D-SS corresponds to the standard Smagorinsky model, X3D-DS to its dynamic version and X3D-iSVV to the implicit approach. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

in Fig. 2(a). Each decomposition (in the  $x$ ,  $y$  and  $z$  axes respectively) allows independent computation of derivatives/interpolations/filtering. Global transpositions to switch from one pencil to another are performed with the MPI command `MPI_ALLTOALL(V)`. The 2D Decomp & FFT library also provides a parallel I/O module to help Xcompact3D handle large data, using MPI-IO. More details about the parallel computing strategy implemented in Xcompact3D can be found in [2].

The scalability of Xcompact3D has been tested recently on various Tier-1/0 supercomputers [ARCHER](#), [MareNostrum](#) and [HazelHen](#) with simulation sizes of more than 68 billion mesh nodes. The scalability plots have been obtained for the Taylor–Green case (see details in the next section) with a very high Reynolds number of 10,000 in a DNS set-up. Xcompact3D exhibits very good strong (see Fig. 2(b)) and weak (see Fig. 2(c)) scaling properties across the three HPC platforms and for up to 100,000 computational cores (with one MPI-process per core). Despite the large number of global communications (up to 70 per time step), the scalability is excellent, except maybe when the number of mesh nodes per core is too small, and the cost of the small communications dominates the calculation.

### 3. Illustrative examples

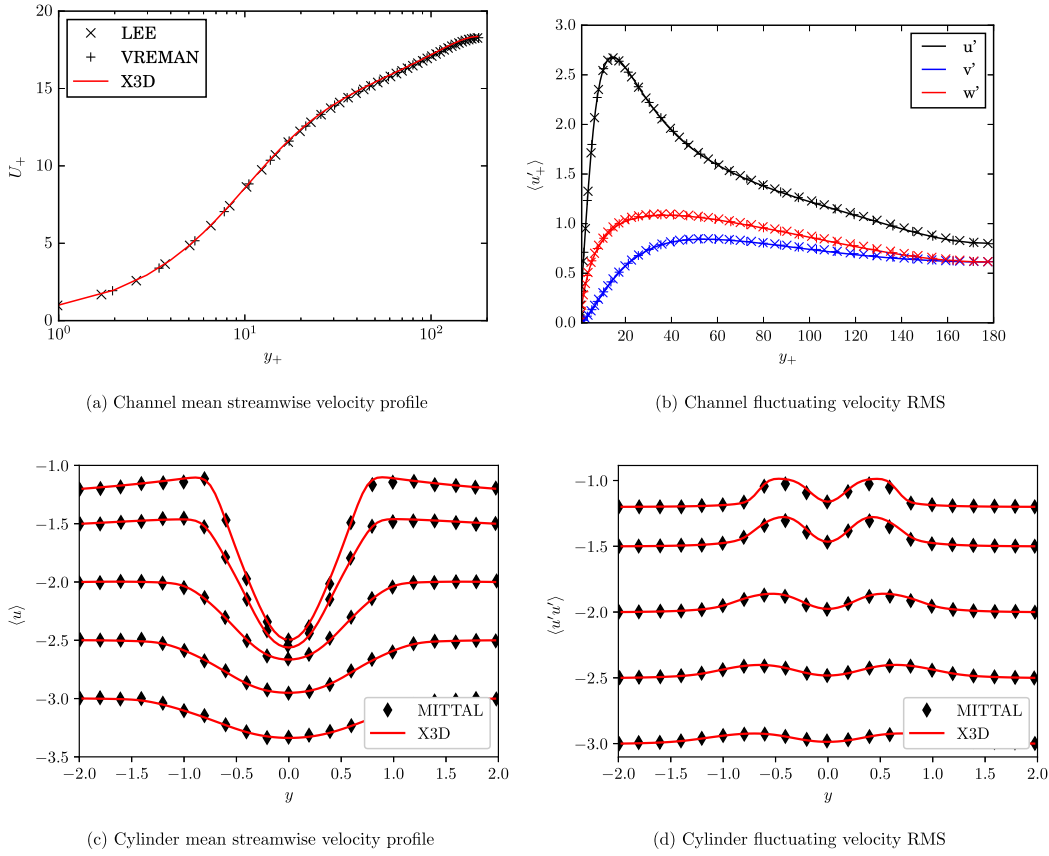
Example applications are defined in the Xcompact3D source under the `examples/` directory. These demonstrate the use of different aspects of the framework and allow new users to begin using Xcompact3D immediately, even if they are not familiar with the framework and its numerical methods. In this section, we present four of these examples to illustrate the use of Xcompact3D to solve well-known CFD benchmarks. Each case consists

of run-time settings specified in `examples/<CASE>/input.i3d` which are easily modifiable by the user and Fortran code implementing case-specific boundary/initial conditions, source terms *etc.* in `src/BC-<CASE>.f90` which must be compiled. Note that a special case `USER` is supplied for users to create their own setups. The input files for each case described can be found in the source repository or can be downloaded [here](#) with the supporting data used to generate the figures in this section. Please note if a user does not know how to define a parameter, default values are defined in `src/parameters.f90` so it should not be a problem to run a simulation even if some parameters are not defined in the input file.

#### 3.1. Taylor–Green vortex

The Taylor–Green Vortex (TGV) is a well-known benchmark in CFD, modelling the transition from an initially laminar state to a fully turbulent one [10,24,25]. It is attractive as a test case due to its simple setup, with the possibility to use a combination of periodic and free-slip boundary conditions. Further details of the case setup can be found in the corresponding `input.i3d` files. The DNS case was performed on a CPU-based cluster at Imperial College London (see performance [here](#)) using 32 cores for 30 min (20,000 time steps) while the LES cases were performed on 64 cores on the same cluster for about 2 h (40,000 time steps). Quantitative comparisons of the temporal evolution of kinetic energy and its dissipation with the reference data of [10] are presented in Fig. 3. For the DNS case at  $Re = 1,600$  the 6th order compact schemes show excellent agreement with the results of [10], whereas the 2nd order schemes show some discrepancies.





**Fig. 4.** Comparison of velocity statistics for channel flow at  $Re_\tau = 180$  with reference data of [19] ( $\times$ ) and [20] (+) - top row; and for flow past a cylinder at  $Re = 300$  at several downstream stations compared with reference data of [21] ( $\diamond$ ) - bottom row.

It clearly demonstrates the ability of high-order schemes to capture a wider range of turbulent scales at a given spatial resolution by comparison to more conventional second-order schemes. The LES results at  $Re = 10,000$  show that the implicit model is able to reproduce the main features of the flow as already reported in [10]. The explicit LES models however struggle to reproduce the DNS data, especially for the kinetic energy dissipation. Note finally that the explicit LES results are different from the ones obtained in [10] (different constant and version for the models).

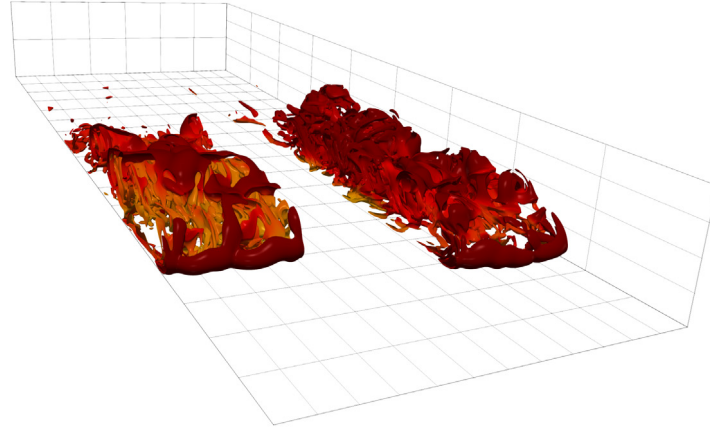
### 3.2. Channel flow and flow past a cylinder

The canonical turbulent channel flow and the flow past a cylinder are very well-documented CFD benchmarks, widely studied theoretically, experimentally and numerically. The main challenge associated with these flows from a simulation point of view is related to the presence of a wall in the computational domain, either at the boundaries for the channel flow or immersed inside the computational domain for the cylinder flow. To demonstrate Xcompact3D's capabilities to handle both wall-bounded flows and flows with immersed boundaries, DNS of a turbulent channel flow at  $Re_\tau = 180$  (based on the half-height of the channel and the friction velocity at the wall) and of a flow past a circular cylinder at  $Re = 300$  (based on the diameter  $D$  of the cylinder and the free stream velocity) are presented and compared with reference data. The numerical parameters for these two simulations can be found in the relevant input file. The channel case was performed on ARCHER with 512 computational cores for 6 h (300,000 time steps) while the cylinder case was performed on ARCHER with 512 computational cores for 60 min (160,000 time steps). Some results are shown in Fig. 4. It can be seen that a very good agreement is obtained with the reference data for the

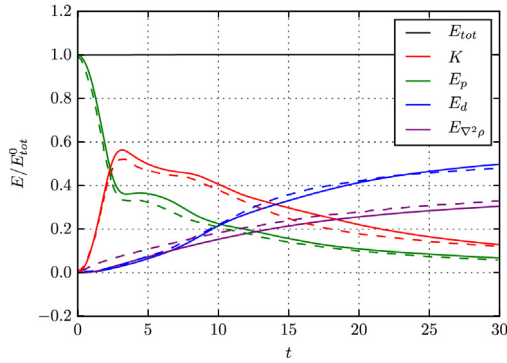
first and second-order moments for the channel flow, and for the mean and fluctuating streamwise components of the velocity downstream of the circular cylinder.

### 3.3. Lock-exchange flow

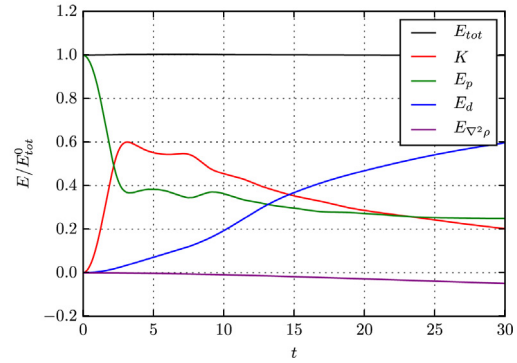
Finally, as an example of the variable density flow capability for Xcompact3D, the well-known lock-exchange case for gravity currents is studied, and data are compared with the data of [23]. The simulations were performed on Imperial's cluster with 512 computational cores for 12 h (100,000 time steps). In this setup, uniformly suspended particle sediments are enclosed in a small portion of the domain separated by a gate with clear fluid. When the gate is removed the particle-fluid mixture flows due to gravity with a mutual inverse interaction between the "heavy" particle-mixture flow and the "light" clear fluid. An illustration of the flow can be seen in Fig. 5(a) where the gravity currents develop a highly 3D turbulence when evolving away from their initial reservoir. When studying buoyancy-driven flows, such as gravity currents, the Boussinesq approximation is frequently used allowing the use of an incompressible formulation only when the density variations are small, typically less than 1%. When changes in density are higher, the low Mach number solver needs to be used (non-Boussinesq case). The reference data of [23] is used to validate the setup in the Boussinesq limit, and a good comparison of energy budgets is found as shown in Fig. 5(b). With limited data for comparison, the energy budget for the non-Boussinesq case is presented with density ratio  $\rho_2/\rho_1 = 0.7$  in Fig. 5(c), noting that the total energy is conserved.



(a) Turbulent structures of the gravity currents (left, Boussinesq case and right, non-Boussinesq case) visualised by isosurfaces of Q-criterion (for the isovalue 1) at  $t=25$ .



(b) Boussinesq energy budget.



(c) Non-Boussinesq energy budget.

**Fig. 5.** 3D snapshots of the gravity currents and evolution of energy budgets for Boussinesq and non-Boussinesq cases, components of the energy budgets are indicated by colour (more details can be found in [22]). The Boussinesq energy budget (solid lines: present results) is compared against the data of [23] (dashed lines).

#### 4. Impact

Xcompact3D is dedicated to DNS/LES, combining the versatility of industrial codes with the accuracy of the best academic codes based on spectral methods (the most accurate ones). It has been used recently worldwide for a variety of projects ranging from the study of the main features of gravity currents [22,26,27], a performance analysis of active and passive drag reduction techniques [28–30], the heat transfer features of a turbulent impinging jet [31,32], the flow over a periodic hill to design a machine learning framework [33,34], active control of turbulent jets [13,35], and nonequilibrium scalings of turbulent wakes [36–38]. Two impact case studies are highlighted in more detail below.

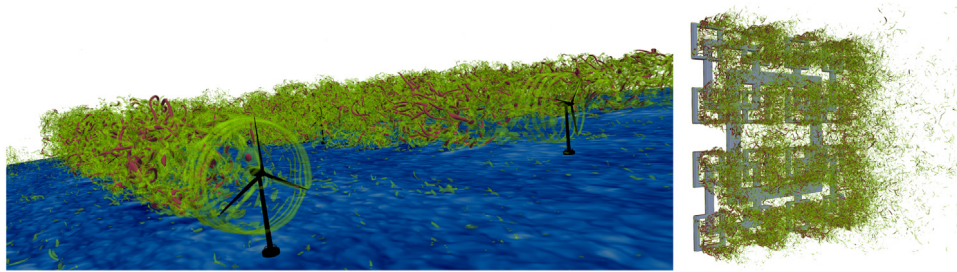
##### 4.1. Fractal-generated turbulence

Laboratory and computational works have recently used multiscale/fractal objects to generate turbulence and have shown that complex multiscale boundary/initial conditions can drastically influence the behaviour of a turbulent flow [39–43]. Fractal geometry is a concept where a given pattern is repeated and split into parts, each being a reduced-copy of the whole, see Fig. 6 for a fractal square grid. Multiscale objects can be designed to be immersed in any fluid flow where there is a need to passively control and design the turbulence generated by the object. They

offer the opportunity to discover new complex flow effects that can help understand how to manage complex fluid flows. The first DNS of the flow generated by fractal objects were performed with Xcompact3D nearly ten years ago [44,45] and have allowed the discovery of the Space Scale Unfolding (SSU) mechanism as a way to dramatically improve stirring in turbulent flows [46,47].

##### 4.2. Wind farm simulator

Modern wind farms consist of multiple turbines clustered together to achieve large-scale energy extraction from a given site. Turbine–turbine interactions through wake interference, enhanced by a complex terrain and atmospheric turbulence phenomena, can lead to reductions of the wind farm energy production by 20% [49]. To this end, a Wind Farm Simulator (WInc3d), built on Xcompact3D, has been developed in order to investigate such turbine–turbine interactions by conducting high-fidelity, turbulence-resolving simulations of the turbine wakes and the ambient atmospheric turbulence [48,50] (see Fig. 6). Additionally, it has also been demonstrated that the implicit LES approach implemented in Xcompact3D is perfectly capable of reproducing the main features of laboratory-scale turbine wakes, with the possibility to investigate various control strategies and operational scenarios [50].



**Fig. 6.** (Left) Wake-to-wake interactions in a virtual wind farm [48]. (Right) Turbulent flow generated downstream of a fractal square grid [47].

## 5. Conclusions and future direction

In this paper, we have shown how Xcompact3D can easily be used to simulate a wide range of turbulent flows. This original framework can be used on a laptop for simple flow configurations but can also be used on the most powerful CPU-based supercomputers in the world for large-scale simulations. Its user-friendliness, simplicity, versatility, accuracy, scalability, portability and efficiency makes it a very attractive tool for the CFD community. The next planned major development of the framework is to port Xcompact3D to GPU-based supercomputers. In addition, we are currently working on a free surface solver to simulate air–water interfaces and on implementing the full compressible Navier–Stokes equations to study compressible turbulent flows.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors would like to thank EPCC for funding part of this work [grant numbers eCSE10-02/eCSE13-03]. Computing time was provided on ARCHER by the UK turbulence consortium [grant number EP/L000261/1 and EP/R029326/1], on MareNostrum and HAZEL HEN by PRACE [grant numbers 2010PA5024 and 2018184381] and on Curie by TGCC under the allocation 2014/2015-2a0912 made by GENCI. Additional computing time was provided by the Imperial College Research Computing Service and by the high-performance facility LAD at the Pontifical Catholic University of Rio Grande do Sul. FS was supported by Petrobras S.A. and financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 88887.154060/2017-00. GD would like to acknowledge funding from EPSRC, UK [grant number EP/R007470/1]. Finally, the authors would like to thank Prof. Christos Vassilicos and Prof. Jorge Silvestrini for their invaluable financial and scientific support in the last 15 years.

## References

- [1] Laizet S, Lamballais E. High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *J Comput Phys* 2009;228(16):5989–6015.
- [2] Laizet S, Li N. Incompact3d: A powerful tool to tackle turbulence problems with up to  $O(10^5)$  computational cores. *Internat J Numer Methods Fluids* 2011;67(11):1735–57.
- [3] McMurtry PA, Jou W-H, Riley J, Metcalfe RW. Direct numerical simulations of a reacting mixing layer with chemical heat release. *AIAA J* 1986;24(6):962–70.
- [4] Golanski F, Fortuné V, Lamballais E. Noise radiated by a non-isothermal, temporal mixing layer. *Theor Comput Fluid Dyn* 2005;19(6):391–416.
- [5] Bartholomew P, Laizet S. A new highly scalable, high-order accurate framework for variable-density flows: Application to non-Boussinesq gravity currents. *Comput Phys Comm* 2019;83–94.
- [6] Li N, Laizet S. 2DECOMP & FFT: a highly scalable 2d decomposition library and FFT interface. In: Cray user group 2010 conference; 2010. p. 1–13.
- [7] Chorin AJ. The numerical solution of the Navier–Stokes equations for an incompressible fluid. *Bull Amer Math Soc* 1967;73(6):928–31.
- [8] Chorin AJ. Numerical solution of the Navier–Stokes equations. *Math Comput* 1968;22(104):745–62.
- [9] Lamballais E, Fortuné V, Laizet S. Straightforward high-order numerical dissipation via the viscous term for direct and large eddy simulation. *J Comput Phys* 2011;230(9):3270–5.
- [10] Dairay T, Lamballais E, Laizet S, Vassilicos JC. Numerical dissipation vs. subgrid-scale modelling for large eddy simulation. *J Comput Phys* 2017;337:252–74.
- [11] Pope SB. *Turbulent flows*. IOP Publishing; 2001.
- [12] Sagaut P. *Large eddy simulation for incompressible flows: an introduction*. Springer Science & Business Media; 2006.
- [13] Gautier R, Laizet S, Lamballais E. A DNS study of jet control with microjets using an immersed boundary method. *Int J Comput Fluid Dyn* 2014;28(6–10):393–410.
- [14] Peskin CS. The immersed boundary method. *Acta Numer* 2002;11:479–517.
- [15] Deville MO, Fischer PF, Fischer PF, Mund E, et al. *High-order methods for incompressible fluid flow*, vol. 9. Cambridge University Press; 2002.
- [16] Karniadakis G, Sherwin S. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press; 2013.
- [17] Lele SK. Compact finite difference schemes with spectral-like resolution. *J Comput Phys* 1992;103(1):16–42.
- [18] Schumann U, Sweet RA. A direct method for the solution of Poisson's equation with Neumann boundary conditions on a staggered grid of arbitrary size. *J Comput Phys* 1976;20(2):171–82.
- [19] Lee M, Moser RD. Direct numerical simulation of turbulent channel flow up to  $Re_\tau \approx 5200$ . *J Fluid Mech* 2015;774:395–415.
- [20] Vreman AW, Kuerten JGM. Comparison of direct numerical simulation databases of turbulent channel flow at  $Re_\tau = 180$ . *Phys Fluids* 2014;26(1):015102.
- [21] Mittal R, Balachandar S. On the inclusion of three-dimensional effects in simulations of two-dimensional bluff-body wake flows. In: *ASME fluids engineering division summer meeting*; 1997. p. 1–6.
- [22] Espath L, Pinto L, Laizet S, Silvestrini J. Two-and three-dimensional direct numerical simulation of particle-laden gravity currents. *Comput Geosci* 2014;63:9–16.
- [23] Necker F, Härtel C, Kleiser L, Meiburg E. High-resolution simulations of particle-driven gravity currents. *Int J Multiph Flow* 2002;28(2):279–300.
- [24] Brachet ME, Meiron DI, Orszag SA, Nickel BG, Morf RH, Frisch U. Small-scale structure of the Taylor–Green vortex. *J Fluid Mech* 1983;130(–1):411–52.
- [25] Wang Z, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, Deconinck H, Hartmann R, Hillewaert K, Huynh HT, Kroll N, May G, Persson P-O, van Leer B, Visbal M. High-order CFD methods: Current status and perspective. *Internat J Numer Methods Fluids* 2013;72(8):811–45.
- [26] Schuch FN, Pinto LC, Silvestrini JH, Laizet S. Three-dimensional turbulence-resolving simulations of the plunge phenomenon in a tilted channel. *J Geophys Res Oceans* 2018;123(7):4820–32.
- [27] Lucchese LV, Monteiro LR, Schettini EBC, Silvestrini JH. Direct numerical simulations of turbidity currents with evolutive deposit method, considering topography updates during the simulation. *Comput Geosci* 2019;133:104306.
- [28] Mahfoze O, Laizet S. Skin-friction drag reduction in a channel flow with streamwise-aligned plasma actuators. *Int J Heat Fluid Flow* 2017;66:83–94.
- [29] Yao J, Chen X, Hussain F. Drag control in wall-bounded turbulent flows via spanwise opposed wall-jet forcing. *J Fluid Mech* 2018;852:678–709.
- [30] Mahfoze O, Moody A, Wynn A, Whalley R, Laizet S. Reducing the skin-friction drag of a turbulent boundary-layer flow with low-amplitude wall-normal blowing within a Bayesian optimization framework. *Phys Rev Fluids* 2019;4(9):094601.



- [31] Dairay T, Fortuné V, Lamballais E, Brizzi L. LES of a turbulent jet impinging on a heated wall using high-order numerical schemes. *Int J Heat Fluid Flow* 2014;50:177–87.
- [32] Dairay T, Fortuné V, Lamballais E, Brizzi L-E. Direct numerical simulation of a turbulent jet impinging on a heated wall. *J Fluid Mech* 2015;764:362–94.
- [33] Wu J-L, Sun R, Laizet S, Xiao H. Representation of stress tensor perturbations with application in machine-learning-assisted turbulence modeling. *Comput Methods Appl Mech Engrg* 2019;346:707–26.
- [34] Wu J, Xiao H, Sun R, Wang Q. Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned. *J Fluid Mech* 2019;869:553–86.
- [35] Ioannou V, Laizet S. Numerical investigation of plasma-controlled turbulent jets for mixing enhancement. *Int J Heat Fluid Flow* 2018;70:193–205.
- [36] Dairay T, Oblgado M, Vassilicos JC. Non-equilibrium scaling laws in axisymmetric turbulent wakes. *J Fluid Mech* 2015;781:166–95.
- [37] Oblgado M, Dairay T, Vassilicos JC. Nonequilibrium scalings of turbulent wakes. *Phys Rev Fluids* 2016;1(4):044409.
- [38] Zhou Y, Vassilicos J. Related self-similar statistics of the turbulent/non-turbulent interface and the turbulence dissipation. *J Fluid Mech* 2017;821:440–57.
- [39] Hurst D, Vassilicos J. Scalings and decay of fractal-generated turbulence. *Phys Fluids* 2007;19(3):035103.
- [40] Seoud R, Vassilicos J. Dissipation and decay of fractal-generated turbulence. *Phys Fluids* 2007;19(10):105108.
- [41] Mazellier N, Vassilicos J. Turbulence without Richardson–Kolmogorov cascade. *Phys Fluids* 2010;22(7):075101.
- [42] Valente P, Vassilicos JC. The decay of turbulence generated by a class of multiscale grids. *J Fluid Mech* 2011;687:300–40.
- [43] Gomes-Fernandes R, Ganapathisubramani B, Vassilicos J. Particle image velocimetry study of fractal-generated turbulence. *J Fluid Mech* 2012;711:306–36.
- [44] Laizet S, Vassilicos J. Multiscale generation of turbulence. *J Multiscale Model* 2009;1(01):177–96.
- [45] Laizet S, Lamballais E, Vassilicos J. A numerical strategy to combine high-order schemes, complex geometry and parallel computing for high resolution DNS of fractal generated turbulence. *Comput & Fluids* 2010;39(3):471–84.
- [46] Laizet S, Vassilicos J. Fractal space-scale unfolding mechanism for energy-efficient turbulent mixing. *Phys Rev E* 2012;86(4):046302.
- [47] Laizet S, Vassilicos J. Stirring and scalar transfer by grid-generated turbulence in the presence of a mean scalar gradient. *J Fluid Mech* 2015;764:52–75.
- [48] Deskos G, Laizet S, Palacios R. WInc3D: A novel framework for turbulence-resolving simulations of wind farm wake interactions. *Wind Energy* 2020;23(3):779–94.
- [49] Barthelmie RJ, Pryor SC, Frandsen ST, Hansen KS, Schepers JG, Rados K, Schlez W, Neubert A, Jensen LE, Neckelmann S. Quantifying the impact of wind turbine wakes on power output at offshore wind farms. *J Atmos Ocean Technol* 2010;27(8):1302–17.
- [50] Deskos G, Laizet S, Piggott MD. Turbulence-resolving simulations of wind turbine wakes. *Renew Energy* 2019;134:989–1002.